DATA HANDLING IN IPSEC ENABLED NETWORK STACK

FIELD OF THE INVENTION

The present invention relates to virtual private networks, more particularly it relates to the processing of data packets with a protocol stack.

BACKGROUND OF THE INVENTION

Over the last few years, the demand to provide networked communications has increased dramatically, and has resulted in enterprises of all sizes providing secure and reliable network access to mobile employees and telecommuters. As the costs of maintaining direct dial-up connections via modem pools and providing a private network infrastructure have grown substantially, a more cost-effective solution has been to use the Internet connections and virtual private network (VPN) servers. A VPN allows a remote client to connect to a corporate network by going directly through any public network, such as the Internet.

One of the technologies that facilitate a VPN is IP Security Architecture (IPSec), which offers an interoperable and open standard for building security into any Internet application. The primary services provided to the IP data packet by IPSec are data confidentiality and authentication. Confidentiality ensures that the data portion of the IP packet is unreadable by unauthorized entities, and the authentication service allows the recipient to be sure that the packet actually comes from the host identified by the source IP address. Both the authentication and confidentiality services are achieved through the use of cryptographic techniques.

The IPSec specification (found in RFC 2401) states that there are several ways to implement IPSec in a host or in conjunction with a router or firewall. The first method is to integrate IPSec into the native IP stack of the operating system, the second method commonly referred to as "Bump in the Stack" (BITS) involves implementing IPSec "beneath" the IP stack and above the network drivers, while the third method known as "Bump in the Wire" (BITW) involves implementing IPSec in a hardware cryptographic processor.

The main advantage of integrating IPSec in the stack and BITS is that such a solution is considerably less expensive than BITW, as they are implemented in software. However, integrating IPSec in the stack requires the source code for the operating system to be available. If the source code is not available then the second method (BITS) is favored. The third method (BITW) is the most expensive implementation, as it requires additional hardware, although such specialized hardware implementations generally provide substantially higher performance in processing cryptographic functions.

However, on some operating systems, such as PALM® OS, it may not be feasible to intercept Internet protocol (IP) packets at the network layer, due to the system architecture. Therefore, the methods described above may not be suitable to implement a driver at the network layer to perform operations on the IP packets.

It is therefore an object of this invention to mitigate at least one of these disadvantages.

SUMMARY OF THE INVENTION

In one of its aspects, the present invention provides a method for providing cryptographic functions to data packets at the data link layer of a network stack. The method includes the steps of intercepting point to point protocol (PPP) datagrams having at least one encapsulated IP packet en route along the protocol stack, decapsulating the PPP datagrams to retrieve the encapsulated IP packet, determining whether to process the IP packet by modifying the IP packet to provide the cryptographic functions, and encapsulating the IP packet for transmission to a next layer of the network stack.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features of the preferred embodiments of the inventor will become apparent in the following defined description in which reference is made to the appended drawings wherein:

Figure 1 shows overview of a system for facilitating a method for implementing security rules and policies within a protocol stack;

Figure 2 shows a communication protocol stack for a handheld device operating system;

Figure 3 shows a block diagram of an IPSec processing module;

Figure 4 shows a flow diagram outlining the steps for intercepting a PPP packet and the steps for modifying the IP packet; and

Figure 5 shows operations performed on the data packet at each step of the flow diagram of Figure 4.

DESCRIPTION OF THE INVENTION

Reference is first made to Figure 1, which is an overview of a system for facilitating a method for implementing security rules and policies within a protocol stack, shown generally by the numeral 10, in a preferred embodiment. The system 10 includes correspondents 12 and 14 communicatively coupled each other, via a communications network 16. It will be appreciated by persons skilled in the art that any network such as a local area network (LAN), a wide area network (WAN), the Internet or a wireless system using, for example, a wireless application protocol (WAP), may be used. The correspondents 12 and 14 are typically computing devices that are, but not limited to, personal computers, handheld devices, cell phones, pagers and microprocessor-based wireless information devices.

The correspondents 12 and 14 include a processing unit, computer readable medium including ROM, flash memory, non-volatile RAM, magnetic disk, optical disk, IC memory card or magnetic tape. Also, the correspondents 12 and 14 execute an operating system such as Microsoft® Windows 2000,Windows CE, UNIX, EPOC, Pocket® PC OS or Palm OS®.

In the preferred embodiment, the correspondent 12 is a handheld device such as Palm or Handspring Visor executing the PALM OS operating system, from Palm Inc, California, USA. Looking at Figure 2, showing the network protocols in the PALM OS environment, the protocol stack 18 is based on the 7-layer OSI model. Thus the stack includes an applications layer 20 for applications such as web browsers and other application programs, a network library 22 coupled to the applications layer via a network library application programming interface (API). The network library 22 includes a transport (TCP and UDP) layer 24, a network (IP) layer 26 and a data link (PPP and SLIP) layer 28. These layers 24, 26 and 28 are integrated to substantially optimize performance, such as speed and space, especially in a handheld environment. Below the network library 22 is a serial library 30 coupled to network library 22 by a serial manager API and communicating with communication hardware 32 on the physical layer. The communication hardware 32 supports a number of communications protocols such as RS232 or X.21 for a serial port on cradle, USB port on cradle. The protocol stack 18 and the application programs may be stored in the computer readable medium or may be embedded in the computer readable medium.

Now referring to Figures 2 and 3, as mentioned above IPSec is usually implemented by adding security at the network layer (IP) and thus enabling security for data via public networks, such as the Internet, by setting up a virtual private network (VPN). IPsec uses an Authentication Header (AH) and an Encapsulating Security Payload (ESP) to apply security to IP packets. The AH and ESP headers include a Security Parameter Index (SPI). The SPI, along with the security protocol in use (AH or ESP) and destination IP address selectors, such as destination IP address or transport layer ports, combine to form the Security Association (SA).

At the sending correspondent 12, there is provided an IPSec security module 34 to implement security on the IP packet. The IPsec module 34 includes a packet interceptor 36 to intercept PPP datagrams and to decapsulate the PPP datagrams to retrieve the encapsulated IP packets. The packet interceptor 36 may be a software module such as a driver placed below the PPP layer of a network stack. The IPsec module 34 determines the type of security to apply to the IP packets by referencing a security policy manager 38. The sending correspondent 12 determines what policy is appropriate for each IP packet, depending on various selectors (for example, destination IP address or transport layer ports), by looking in the security policy manager 38, which indicates the relevant policy for any particular packet. The packet either requires IPsec processing of some sort, in which case it is passed to an IPsec processing module 40 for processing; or it does not, in which case it is simply passed along for normal IP processing. The IPsec processing module 40 performs packet-per packet processing by examining the packets in order to select and apply cryptographic transformations on the IP packets as known the art. In instances where processing is not required, the IP packets may be dropped or the IP packets proceed up or down the protocol stack 18. Outbound packets are checked against the security policy manager 38 to see what kind (if any) of IPsec processing to apply, while inbound packets are checked against the security policy manager 38 to see what kind of IPsec service should be present in those IP packets.

After processing the IP packets are encapsulated to form a new PPP datagram, generally PPP uses the High-Level Data Link Control (HDLC) protocol as a basis for encapsulating datagrams and provides framing of packets transmitted over bit- oriented synchronous links. Also, the packet interceptor 36 monitors the Link Control Protocol (LCP) packets for ACCM parameters for HDLC framing for each direction. PPP uses LCP to establish, configure, and test the data-link connection, and network control protocols (NCP) for establishing and configuring different network-layer protocols.

The process for applying cryptographic functions to IP packets at the PPP layer of a network stack is illustrated by a flow chart of Figure 4, in conjunction with Figure 5. The PPP datagram 42 includes the following frame fields: a flag field which indicates the beginning or end of a frame and consists of the binary sequence, an address field, a control field, a protocol field to identify the protocol encapsulated in the data field of the frame, a data field that contains the datagram for the protocol specified in the protocol field and a frame check sequence (FCS) for error detection. The process starts with step 100 where a byte stream in the form of a plurality of PPP datagrams 42 is intercepted en route along the protocol stack 18. In step 102, the PPP datagram 42 is decapsulated to retrieve the encapsulated IP packet 44, and then a determination 104 is performed as to how the IP packet 44 should be processed. Should the IP packet 44 require processing, it is transformed by adding cryptographic functions to the IP packets 44 in step 106 resulting in a tunnel mode protected packet or a transport modc protected packet 46, or else the IP packet 44 is not processed 108. In step 110, the IP packet is encapsulated with a PPP header and trailer and a new PPP datagram 48 and the modified IP packet is thus formed and transmitted to the physical interface.

Since encapsulation results in the original IP packet 44 being hidden or included inside a PPP datagram 48, the IP header of the tunnel mode protected packet 46 provides the necessary routing information, enabling the packet 44 to travel through the communication network 12 without revealing the final destination stored in the original IP packet header. Once the encapsulated IP packets 44 reach their destination, the encapsulation header is removed and the original IP packet header is used to route the packet to its final destination.

The above-described embodiments of the invention are intended to be examples of the present invention and alterations and modifications may be effected thereto, by those of skill in the art, without departing from the scope of the invention which is defined solely by the claims appended hereto.